



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/810,716	03/16/2001	Hiang-Swee Chiang	3996-4002US1	2184
23377	7590	05/17/2006	EXAMINER	
WOODCOCK WASHBURN LLP ONE LIBERTY PLACE, 46TH FLOOR 1650 MARKET STREET PHILADELPHIA, PA 19103			WOOD, WILLIAM H	
			ART UNIT	PAPER NUMBER
			2193	

DATE MAILED: 05/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/810,716	Applicant(s) CHIANG, HIANG-SWEE	
	Examiner William H. Wood	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 February 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 2, 4-29, 31-50, 52-64, 66-78 and 162-169 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 2, 4-29, 31-50, 52-64, 66-78 and 162-169 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claims 2, 4-29, 31-50, 52-64, 66-78 and 162-169 are pending and have been examined.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 2, 4-13, 15, 18-20, 22-29, 31-37, 39, 42-50, 52-, 57, 59, 63, 64, 66-70, 72, 75-78 and 162-169 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lau** (USPN 5,987,247) in view of **Lindhorst** et al. (USPN 6,337,696) in further view of **Quaeler-Bock** et al. (USPN 6,023,271).

Claim 2

Lau disclosed a method of generating computer code for a application (*column 5, lines 33-40*), comprising:

generating an application framework code and an event handler skeleton (*column 6, lines 24-29; column 5, lines 33-40; column 13, lines 28-44; column 5, lines 33-40*),

receiving application business logic objects (*column 6, lines 24-29*);

receiving methods (*column 13, lines 28-44*);

organizing the application framework code, the application business logic objects and the event handler methods into application source code (*column 4, lines 25-27, linking and compiling, along with building and preparing the code*);

Lau did not explicitly state generating code for a *web* application. **Lindhorst** demonstrated that it was known at the time of invention to generate code for web applications (column 2, lines 11-19; column 3, lines 1-4; and column 4, lines 10-16) using, among other elements, a graphical interface input file (column 11, lines 37-40). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the application framework generation system of **Lau** with graphical design input for the web as found in **Lindhorst**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide less technical and thus easier methods, such as frameworks and automatic code generation, for average users to program in various known environments, like the web (**Lindhorst**: column 3, lines 37-45; **Lau**: column 2, lines 43-47).

Lau did not explicitly state *receiving event handler methods; and wherein generating an event handler skeleton further comprises: parsing at least one input file* (**Lindhorst**: column 11, lines 37-40); *reviewing the parsed input file for a tag type, an attribute name and an attribute value* (**Lindhorst**: column 13,

*lines 22-64); and determining an event handler method based on the tag type, the attribute name and the attribute value (**Lindhorst**: column 13, lines 22-64).*

Lindhorst demonstrated that it was known at the time of invention to provide event handler methods (column 3, lines 16-20; column 11, line 66 to column 12, line 17; figure 6) and generating/producing them in the manner cited above. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code generation system of **Lau** with provided event handler methods as found in **Lindhorst**'s teaching. Further, it would have been obvious to one of ordinary skill in the art at the time of invention to implement the generation of event handler methods as described above through **Lindhorst**. This implementation would have been obvious because one of ordinary skill in the art would be motivated to free a user from being required to know complex technical details of programming, thus making it easier (**Lindhorst**: column 3, lines 38-45).

Lau and **Lindhorst** did not explicitly state *compiling/binding the web application source code with input files at runtime or receive GUI input files*.

Quaeler-Bock demonstrated it was known at the time of invention to bind at runtime GUI components (input files) to business objects (source code) (column 3, lines 21-25) and thus receiving GUI input files. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code development system of **Lau** and **Lindhorst** with run-time binding of various

files/inputs as suggested by **Quaeler-Bock**'s teachings. This implementation would have been obvious because one of ordinary skill in the art would be motivated to reduce error-prone operations during code development (column 3, lines 5-10). Finally, runtime interpretation/binding indicates allowing updated/modified files up to runtime (ie. "synchronization").

Claim 4

Lau and **Lindhorst** disclosed the method of claim 2, wherein the web application source code is generated in an object-oriented programming language (**Lau**: column 6, line 34).

Claim 5

Lau and **Lindhorst** disclosed the method of claim 4, wherein the object-oriented programming language is Java (column 6, line 34).

Claim 6

Lau and **Lindhorst** disclosed the method of claim 4, wherein the object-oriented programming language is C++ (column 3, line 65).

Claim 7

Lau and **Lindhorst** disclosed the method of claim 2, further comprising determining if the application framework code is available for the web

application (**Lau:** column 6, lines 21-22; and column 5, lines 46-50; must determine if saved framework exists for future editing/changing).

Claim 8

Lau and **Lindhorst** disclosed the method of claim 2, further comprising generating a business logic foundation code (**Lau:** column 6, lines 24-29).

Claim 9

Lau and **Lindhorst** disclosed the method of claim 2, further comprising generating a graphical user interface code (**Lau:** column 5, line 39).

Claim 10

Lau and **Lindhorst** disclosed the method of claim 9, wherein generating a graphical user interface code is based on the input files (**Lau:** column 5, lines 33-39; design; column 4, lines 11-28).

Claim 11

Lau and **Lindhorst** disclosed the method of claim 2, wherein generating an event handler skeleton is based on the input files (**Lau:** column 5, lines 33-39; design; column 4, lines 11-28).

Claim 12

Lau and **Lindhorst** disclosed the method of claim 2, further comprising compiling the web application source code (**Lau**: column 4, lines 25-27).

Claim 13

Lau and **Lindhorst** did not explicitly state the method of claim 2, further comprising interpreting the web application source code. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a interpretation system. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code development system of **Lau** and **Lindhorst** with interpreting code as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a system of easy programmability (interpretation can be changed quickly on the fly, especially useful in system testing).

Claim 15

Lau and **Lindhorst** disclosed the method of claim 2, wherein the input files are in HTML format (**Lindhorst**: column 11, lines 37-40).

Claim 18

Lau and **Lindhorst** disclosed the method of claim 2, further comprising receiving modified input files (*see motivation under claim 2; runtime interpretation/binding indicates allowing updated/ modified files up to runtime*).

Claim 19

Lau and **Lindhorst** did not explicitly state the method of claim 18, further comprising compiling the modified input files at runtime. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a just-in-time compiling system (in other words compiling at run time). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of **Lau** and **Lindhorst** with run-time compiling as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for changes/improvements right up until actual use of code (see claim 18).

Claim 20

Lau and **Lindhorst** disclosed the method of claim 19, further comprising binding the web application source code with the modified input files at runtime (see claim 2 above).

Claim 22

Lau and **Lindhorst** did not explicitly state the method of claim 18, further comprising interpreting the modified input files at runtime. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a interpretation system. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the development system of **Lau** and **Lindhorst** with interpreting code as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a system of easy programmability (interpretation can be changed quickly on the fly, especially useful in system testing).

Claim 23

Lau and **Lindhorst** disclosed the method of claim 22, further comprising binding the web application source code with the interpreted modified input files at runtime (*see claim 2 and 22; further binding required in order for code to work correctly*).

Claim 24

Lau and **Lindhorst** disclosed the method of claim 2, further comprising generating application runtime properties (**Lau**: column 5, lines 39-40; attributes at least).

Claim 25

Lau and **Lindhorst** did not explicitly state the method of claim 2, further comprising generating application SQL statements. **Lau** demonstrated that it was known at the time of invention to utilize database management systems in business logic (column 8, lines 16-25). Official Notice is taken that SQL was known at the time of invention. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code framework system of **Lau** with generating SQL as well. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide **Lau**'s system with the ability to communicate with as many differing systems/environments as possible and thus increasing flexibility and usability.

Claim 26

Lau and **Lindhorst** disclosed the method of claim 2, wherein the application framework code comprises an application object and a servlet web application framework object (column 5, lines 15-19).

Claims 27-78 and 162-166

The limitations of system claims 27-166 correspond to the limitations of method claims 2-26 and as such are rejected in the same manner.

Claim 167

Lau, Lindhorst and **Quaeler-Bock** disclosed the method of claim 1, further comprising:

determining if an application framework code is available for the web application (*column 6, lines 21-22; and column 5, lines 46-50; must determine if saved framework exists for future editing/ changing*); and

if the application framework is not available, then generating the application framework code (*column 5, lines 33-40; requiring generation if no saved information is present*).

Claims 168-169

The limitations of claims 168-169 are substantially the same as for claim 167 and as such are rejected in the same manner.

3. Claims 14, 16-17, 21, 38, 40-41, 58, 60-62, 71 and 73-74 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lau** (USPN 5,987,247) in view of **Lindhorst** et al. (USPN 6,337,696) in view of **Quaeler-Bock** et al. (USPN 6,023,271) and in further view of **APA** (Applicant Admitted Prior Art).

Claim 14, 16-17

Lau and **Lindhorst** did not explicitly state the method of claim 2, wherein the input files are in XML, cHTML or WML format. **APA** demonstrated that it was known at the time of invention to utilize XML and WML (page 3, lines 13-14). Official Notice is take that cHTML was known at the time of invention. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the input files of **Lau** and **Lindhorst** with the above formats as found in **APA**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide as many formats as possible in order to be of use to the largest community of developers possible and thus increase usefulness of the system.

Claim 21

Lau and **Lindhorst** did not explicitly state the method of claim 20, wherein the modified input files are compiled into DOM objects at runtime (*APA: page 3, lines 14-16*). **APA** demonstrated that it was known at the time of invention to compile mark up language files into DOM (page 3, lines 14-16). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code development system of **Lau** and **Lindhorst** with DOM compilation as found in **APA**'s teaching. This implementation would have been

obvious because one of ordinary skill in the art would be motivated to provide a easily handled structure for development (**APA**: page 3, lines 16-22).

Claims 38, 40-41, 58, 60-62, 71 and 73-74

The limitations of claims 38, 40-41, 58, 60-62, 71 and 73-74 are substantially the same as for claims 14, 16-17 and 21 and as such are rejected in the same manner.

Response to Arguments

4. Applicant's arguments with respect to claims 1-78 and 162-174 have been considered but are moot in view of the new ground(s) of rejection.

Applicant argues no motivation to combine **Lindhorst** with **Lau** and **Lindhorst** fails to disclose “determining an event handler method based on the tag type, the attribute name and the attribute value”. First, the motivation is supplied as indicated above (This implementation would have been obvious because one of ordinary skill in the art would be motivated to free a user from being required to know complex technical details of programming, thus making it easier (**Lindhorst**: column 3, lines 38-45). Second, under the broadest reasonable interpretation an event handler is determined by “one or more of the tag type, the attribute name and the attribute value”. Refer to Table 2 (**Lindhorst** column 13, lines 30-65), showing “events” and “methods” related to specific “HTML tags”. A user does perform a linking, but that linking is “based

on" certain tags having certain/appropriate events and methods. Further note, Table 2's "HTML tag" demonstrates tag type (for example scriptable tags; column 13, line 24), attribute name (for example FORM) and attribute value (for example "button"). The rejections are maintained as above indicated. Applicant's request for interview in the conclusion of Applicant's remarks of 21 February 2006 is unnecessary in view this response.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Art Unit: 2193

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (571)-272-3736. The examiner can normally be reached 9:00am - 5:30pm Monday thru Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)-272-3719. The fax phone numbers for the organization where this application or proceeding is assigned are (571)273-8300 for regular communications.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR systems, see <http://pair-direct.uspto.gov>. For questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.



William H. Wood
Patent Examiner
AU 2193
May 12, 2006



KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100